AMENDMENTS TO THE CLAIMS

Please amend Claims 2, 10, 31-35, 37, 50, 51, and 53 as indicated below. Also, please cancel Claims 23-29, 44-49, 52 and 54 without prejudice as indicated below.

1. (Canceled)

2. (Currently Amended) A method for providing data transfers between a processor and a component, the method comprising:

routing requests originating from ~~a component to~~ a processor to a component through a target controller and handling requests originating from the processor to the component by;

buffering a first address with a first address buffer in response to a deferred read request originating from the processor to the component, and associating a first bi-directional data buffer with the first address wherein the first bi-directional data buffer is configured to hold a first data value ~~requested by the processor from the component~~ associated with the deferred read request;

setting status information to indicate that the first address buffer is associated with the deferred read request;

buffering a second address with a second address buffer in response to a write request originating from the processor to the component, and associating a second bi-directional data buffer with the second address wherein the second bi-directional data buffer is configured to hold a second data value ~~written by the processor to the component~~ associated with the write request, the first and second address buffers being in communication with the processor and the component~~, wherein the processor operates at a different speed than the component~~;

setting status information to indicate that the second address buffer is associated with the write request;

buffering the first data value requested by the processor with the first bi-directional data buffer when the data is obtained with the deferred read request

from the component and buffering the second data value associated with the write request written by the processor with the second bi-directional data buffer, the first and second bi-directional data buffers being in communication with the processor and the component, wherein the first and second address buffers are separate from the first and second data buffers;

monitoring the first and second address buffers and the first and second data buffers to determine when ~~the first address and~~ the data buffers have obtained the first data value associated with the deferred read request~~requested by the processor~~ and the second data value associated with the write request ~~address and data buffers have written the second data value to the component~~;

controlling the first address buffer and the first bi-directional data buffer as a matched pair such that the first address held in the first address buffer corresponds to the first data value associated with the deferred read request~~requested by the processor from the component~~;

controlling the second address buffer and the second bi-directional data buffer as a matched pair such that the second address held in the second address buffer corresponds to the second data value associated with the write request~~written by the processor to the component~~;

reading the status information from the first address buffer to determine ~~a priority~~ the status of the first data value associated with the deferred read request;

reading status information from the second address buffer to determine the ~~priority~~ status of the second data value associated with the write request; and

controlling the order of bi-directional data flow through the first and second bi-directional data buffers such that data flows between the processor and the component while the processor is processing other instructions and, wherein controlling the order of the bi-directional data flow through the first and second bi-directional data buffers is variable and based on the ~~priority~~ status information

of the first and second data values such that the bi-directional data flow does not occur in a first-in-first out manner.

3. (Previously Presented) The method of Claim 2, wherein the first and second bi-directional data buffers are in communication with the processor via a bus.

4. (Previously Presented) The method of Claim 3, wherein the first and second bi-directional data buffers are in communication with the bus via a bus master controller and a bus slave controller.

5. (Previously Presented) The method of Claim 2, wherein the first address buffer further comprises status bits.

6. (Previously Presented) The method of Claim 5, wherein the status bits relate to the type of request being made by the processor.

7. (Previously Presented) The method of Claim 2, wherein said controlling the first address buffer and the first bi-directional data buffer as a matched pair is performed with pointers.

8. (Canceled)

9. (Previously Presented) The method of Claim 2, wherein said act of controlling bi-directional data flow is performed with at least one input data arbiter.

10. (Currently Amended) A method for controlling data transfers between a processor and a component, the method comprising:

~~routing requests from a component for data from a processor through a target controller and routing requests from the processor for data from the component;~~

~~buffering with a plurality of address buffers, the requests from the component to the processor;~~

~~buffering with a plurality of address buffers, the requests from the processor to the component, wherein the processor operates at a different speed than the component;~~

associating a plurality of bi-directional data buffers with ~~the~~ a plurality of address buffers such that at least one bi-directional data buffer is matched with at least one address buffer for each request, and wherein the bi-directional data buffers are configured to hold data to be obtained from either the component or the processor;

storing status information in each of the plurality of address buffers, the status information identifying whether the address buffers are associated with one or more deferred read requests from a processor to a component and one or more write requests from the processor to the component~~determining the priority status of data transfers associated with the address requests~~;

monitoring the status information in each of the plurality of address buffers ~~and the first~~ to determine when address buffers have completed a task and are available for a further task;

bi-directionally buffering with a plurality of bi-directional data buffers data transfers between the processor and the component, wherein said data transfers can be performed out of a previously defined order based on the ~~priority~~ status information of the deferred read requests and the write requests ~~each of the data transfers and~~ such that data transfers can be performed while the processor is processing other instructions and such that the bi-directional data flow does not occur in a first-in-first out manner; and

controlling ~~said buffering address requests and~~ said bi-directionally buffering through said plurality of bi-directional data buffers such that each of the buffered data transfers relates to an address held in one of the plurality of address buffers.

11. (Previously Presented) The method of Claim 10, additionally comprising indicating which of the plurality of bi-directional data buffers is available to accept new data.

12. (Previously Presented) The method of Claim 11, wherein said act of indicating is performed with reference pointers.

13. (Canceled)

14. (Previously Presented) The method of Claim 10, wherein said act of buffering address requests includes the use of an input arbiter and an output arbiter.

15. (Previously Presented) The method of Claim 10, wherein said act of bi-directionally buffering is performed with an input arbiter and an output arbiter.

16. (Previously Presented) The method of Claim 10, wherein the plurality of address buffers comprises at least three address buffers and wherein the plurality of bi-directional data buffers comprises at least three bi-directional data buffers and wherein each address buffer is matched as a pair with a corresponding data buffer.

17.– 30. (Canceled)

31. (Currently Amended) A method for transferring data between a processor and a component utilizing a plurality of address buffers and a plurality of data buffers, the method comprising:

receiving a first request that originates from the processor, ~~receiving~~ wherein the first request comprises ~~including~~ an associated first address from the processor, and wherein the first request is associated with a first deferred read request;

determining whether at least ~~one of a plurality of~~ a first address ~~buffers~~ buffer and an associated first bi-directional data buffer are available for the first deferred read request, wherein the associated first bi-directional data buffer is configured to buffer the data identified by the first address from the processor;

storing the first address in the ~~at least one~~ first address buffer;

storing status information ~~indicative of a priority of the first request~~ in the ~~at least one~~ first address buffer indicating that the first request is associated with a deferred read request;

receiving a second request that originates from the processor for a second deferred read request while the first deferred read request is pending, wherein the second deferred request comprises a second address;

determining whether a second address buffer and an associated second bi-directional data buffer are available for the second deferred read request, wherein the associated second bi-directional data buffer is configured to buffer the data identified by the second address;

storing the second address in the second address buffer;

storing status information in the second address buffer indicating that the second request is associated with a deferred read request;

~~buffering data identified by the first address with the bi-directional data buffer, wherein the data is obtained from a component and is provided to the processor; and~~

ordering, based on the ~~priority~~ status information in the first address ~~of the first request~~and the status information in the second address, the transmission of the data from the bi-directional data buffer to the processor and such that data flows bi-directionally with processing by the processor of other instructions and such that the bi-directional data flow does not occur in a first-in-first out manner.; ~~and~~

~~receiving a second request that originates from the component, the second request including a second address;~~

~~determining whether at least one of the plurality of address buffers and associated bi-directional data buffers are available, wherein the associated bi-directional data buffer is configured to buffer the data identified by the second address from the component;~~

~~storing the second address in the at least one address buffer; and~~

~~buffering data identified by the second address with the bi-directional data buffer, wherein the data is obtained from the processor and is provided to the component.~~

32. (Currently Amended) The method of Claim 31, additionally comprising receiving the second address into the ~~at least one~~ second address buffer while data is being read from the bi-directional data buffer.

33. (Currently Amended) The method of Claim 31, wherein the ~~at least one~~ first address buffer and the first bi-directional data buffer are in communication with the processor via a bus.

34. (Currently Amended) The method of Claim 33, wherein the ~~at least one~~ first address buffer and the first bi-directional data buffer are in communication with the bus via a bus master controller and a bus slave controller.

35. (Currently Amended) The method of Claim 31, wherein the first bi-directional data buffer and the ~~at least one~~ first address buffer are associated with each other through the use of pointers.

36. (Canceled)

37. (Currently Amended) An apparatus for controlling data transfers between a processor and a component, the apparatus comprising:

means for buffering at least a first address associated with a first request associated with a deferred read request from a processor to a component and buffering at least a second address associated with a second request associated with a write request from ~~a component to a~~ the processor to the component;

means for bi-directionally buffering data transfers between the processor and the component, that are associated with the first and second addresses;

means for storing status information indicative of the first request being associated with the deferred read request and the second request being

associated with a write request, the status information further indicative of a ~~priority~~ status of the buffered data transfers; and

means for controlling the means for buffering and the means for bi-directionally buffering so that each of the buffered data transfers relates to the first and second addresses held in the means for buffering, wherein the means for controlling further coordinates an order of said data transfers based at least on the ~~priority~~ status information of each buffered data transfer and such that data flows bi-directionally with processing by the processor of other instructions such that the bi-directional data flow does not occur in a first-in-first out manner.; ~~and~~

~~means for routing the data transfers from the processor to the component and routing the data transfers from the component to the processor.~~

38. (Canceled)

39. (Previously Presented) The apparatus of Claim 37, wherein the means for buffering includes a plurality of address buffers.

40. (Previously Presented) The apparatus of Claim 37, wherein the means for bi-directionally buffering includes a plurality of data buffers.

41. (Previously Presented) The apparatus of Claim 37, wherein the means for buffering includes an input arbiter and an output arbiter.

42. (Previously Presented) The apparatus of Claim 37, wherein the means for bi-directionally buffering includes an input arbiter and an output arbiter.

43. (Previously Presented) The method of Claim 23, additionally comprising providing signals with an arbiter in communication with said status bits to grant access to the first buffer and to the second buffer such that a third address can be written to the first buffer while data is read from the second buffer.

44.– 49. (Canceled)

50. (Previously Presented) The method of Claim 2, wherein the data flowing concurrently comprises one or more of:

the processor writing data associated with the write request to the ~~first~~ second data buffer at the same time that data associated with the deferred read request is being ~~read from a PCI bus into the second~~ loaded into the first data buffer;

~~a deferred data read from the first data buffer occurring concurrently with a data read from a PCI bus to the second data buffer; and~~

~~performing a deferred data read from the first data buffer to the processor at the same time as performing a data write operation from the second data buffer to a PCI bus~~.

51. (Currently Amended) The method of Claim 10, wherein the concurrent data transfer comprises one or more of:

the processor writing data associated with the wire request to ~~a first~~ the second data buffer at the same time that data associated with the deferred read request is being ~~read from a PCI bus~~ loaded into ~~a second~~ the first data buffer;

~~a deferred data read from a first data buffer occurring concurrently with a data read from a PCI bus to a second data buffer; and~~

~~performing a deferred data read from a first data buffer to the processor at the same time as performing a data write operation from a second data buffer to a PCI bus~~.

52. (Canceled)

53. (Currently Amended) The method of Claim 31, wherein the data flowing concurrently comprises one or more of:

the processor writing data associated with the write request to ~~a first~~ the data buffer at the same time that data associated with the deferred read request is being ~~read from a PCI bus~~ loaded into ~~a second~~ the first data buffer;

~~a deferred data read from a first data buffer occurring concurrently with a data read from a PCI bus to a second data buffer; and~~

~~performing a deferred data read from a first data buffer to the processor at the same time as performing a data write operation from a second data buffer to a PCI bus~~.

54. (Canceled)